2. Ariel

Program Name: Ariel.java **Input File: ariel.dat**

Your friend Ariel is an architectural student studying subway systems, and they need your help with their homework. They need to plan out a subway stop, and they have been given the times that all the trains will be arriving and departing from the stop. Write a program to determine the minimum number of train stops required to ensure that there are no delays, in other words, every train should arrive when there is at least one open stop.

Input: The input will begin with an integer, `n` (`0 < n <= 1000`), denoting the number of test cases to follow. Each test case will consist of two lines of space separated strings denoting the arrival times of all trains on the first line, and departure times of all trains on the second line, all in the format `"H:MM"`, and all minutes will be multiples of 5. The i^{th} index in both lists correspond, as in, all arrival times and departure times in the same index in their respective lists refer to the arrival and departure of the same train. There will never be two trains with the same arrival AND departure time, although trains may share the same arrival OR departure. It can be assumed that the trains operate on a 24-hour cycle, so trains arrive and leave at the same time every day.

Output: Output the integer denoting the minimum number of train platforms required so that the current train schedule will have no delays for trains when arriving. If one train arrives at the same time that another train departs, then you will only need one platform (the train engineers have been specifically trained for these situations at the same school as those two guys from the Polar Express).

Sample input:

```
3
9:30 9:45 9:50 10:30 11:30 12:00
10:00 10:05 10:15 11:00 12:00 12:10
0:00 1:00 2:00 3:00 4:00
0:10 1:10 2:10 3:10 4:10
8:15 8:25 8:30 8:35 8:40 8:45 9:00
8:25 8:40 8:45 8:45 8:55 9:00 9:30
```

Sample output:

```
3
1
3
```